Analysis of a Natural Gradient Algorithm on Monotonic Convex-Quadratic-Composite Functions

Youhei Akimoto

Analysis of a Natural Gradient Algorithm on Monotonic Convex-Quadratic-Composite Functions

Youhei Akimoto
Project TAO INRIA Saclay, LRI Université Paris-Sud, 91405 Orsay Cedex, France
Youhei.Akimoto@lri.fr

ABSTRACT

In this paper we investigate the convergence properties of a variant of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Our study is based on the recent theoretical foundation that the pure rank-μ update CMA-ES performs the natural gradient descent on the parameter space of Gaussian distributions. We derive a novel variant of the natural gradient method where the parameters of the Gaussian distribution are updated along the natural gradient to improve a newly defined function on the parameter space. We study this algorithm on composites of a monotone function with a convex quadratic function. We prove that our algorithm adapts the covariance matrix so that it becomes proportional to the inverse of the Hessian of the original objective function. We also show the speed of covariance matrix adaptation and the speed of convergence of the parameters. We introduce a stochastic algorithm that approximates the natural gradient with finite samples and present some simulated results to evaluate how precisely the stochastic algorithm approximates the deterministic, ideal one under finite samples and to see how similarly our algorithm and the CMA-ES perform.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization, Gradient methods, Unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms, Theory

Keywords

Covariance Matrix Adaptation, Natural Gradient, Hessian Matrix, Information Geometric Optimization, Theory

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES, [13–15]) is a stochastic search algorithm for non-separable and ill-conditioned black-box continuous optimization. In the CMA-ES, search points are generated from a Gaussian distribution and the mean vector and the covariance matrix of the Gaussian distribution are adapted by using the sampled points and their objective value ranking. These parameters' update rules are designed so as to enhance the probability of generating superior points in the next iteration in a way similar to but slightly different from the (weighted) maximum likelihood estimation. Adaptive-ESs including the CMA-ES are successfully applied in practice. However, their theoretical analysis even on a simple function is complicated and linear convergence has been proven only for simple algorithms compared to the CMA-ES [6, 17].

Resent studies [1, 11] demonstrate the link between the parameter update rules in the CMA-ES and the natural gradient method, the latter of which is the steepest ascent/descent method on a Riemannian manifold and is often employed in machine learning [2, 4, 8, 21–23]. The natural gradient view of the CMA-ES has been developed and extended in [5] and the Information-Geometric Optimization (IGO) algorithm has been introduced as the unified framework of natural gradient based stochastic search algorithms. Given a family of probability distributions parameterized by $\theta \in \Theta$, the IGO transforms the original objective function, f, to a fitness function, J, defined on Θ. The IGO algorithm performs a natural gradient ascent aiming at maximizing J. For the family of Gaussian distributions, the IGO algorithm recovers the pure rank-μ update CMA-ES [14], for the family of Bernoulli distributions, PBIL [7] is recovered. The IGO algorithm can be viewed as the deterministic model of a recovered stochastic algorithm in the limit of the number of sample points going to infinity.

The IGO offers a mathematical tool for analyzing the behavior of stochastic algorithms. In this paper, we analyze the behavior of the deterministic model of the pure rank-μ update CMA-ES, which is slightly different from the IGO algorithm. We are interested in knowing what is the target matrix of the covariance matrix update and how fast the covariance matrix learns the target. The CMA is designed to solve ill-conditioned objective function efficiently by adapting the metric—covariance matrix in the CMA-ES—as well as other variable metric methods such as quasi-Newton methods [20]. Speed of optimization depends on the precision and the speed of metric adaptation to a great ex-

tend. There is a lot of empirical evidence that the covariance matrix tends to be proportional to the inverse of the Hessian matrix of the objective function in the CMA-ES. However, it has not been mathematically proven yet. We are also interested in the speed of convergence of the mean vector and the covariance matrix. Convergence of the CMA-ES has not been reported up to this time. We tackle these issues in this work.

In this paper, we derive a novel natural gradient algorithm in a similar way to the IGO algorithm, where the objective function f is transformed to a function J in a different way from the IGO so that we can derive the explicit form of the natural gradient for composite functions of a strictly increasing function and a convex quadratic function. We call the composite functions *monotonic convex-quadratic-composite* functions. The resulting algorithm inherits important properties of the IGO and the CMA-ES, such as invariance under monotone transformation of the objective function and invariance under affine transformation of the search space. We theoretically study this natural gradient method on monotonic convex-quadratic-composite functions. We prove that the covariance matrix adapts to be proportional to the inverse of the Hessian matrix of the objective function. We also investigate the speed of the covariance matrix adaptation and the speed of convergence of the parameters.

The rest of the paper is organized as follows. In Section 2 we propose a novel natural gradient method and present a stochastic algorithm that approximates the natural gradient from finite samples. The basic properties of both algorithms are described. In Section 3 we study the convergence properties of the deterministic algorithm on monotonic convex-quadratic-composite functions. The convergence of the condition number of the product of the covariance matrix and the Hessian matrix of the objective function to one and its linear convergence are proven. Moreover, the rate of convergence of the parameter is shown. In Section 4, we conduct experiments to see how accurately the stochastic algorithm approximates the deterministic algorithm and to see how similarly our algorithm and the CMA-ES behave on a convex quadratic function. Finally, we summarize and conclude this paper in Section 5.

2. THE ALGORITHMS

We first introduce a generic framework of the natural gradient algorithm that includes the IGO algorithm.

The original objective is to minimize $f : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X}$ is a metric space. Let $\mathcal{F}$ and μ be the Borel σ-field and a measure on $\mathbb{X}$. Hereunder, we assume that f is μ-measurable. Let ν represent any monotonically increasing set function on $\mathcal{F}$, i.e., $\nu(A) \leqslant \nu(B)$ for any A, $B \in \mathcal{F}$ s.t. $A \subseteq B$. We transform f to an *invariant cost* function defined as $V_f : x \mapsto \nu[y : f(y) \leqslant f(x)]$. Given a family of probability distributions P_θ on $\mathbb{X}$, we define a *quasi*-objective function J on the parameter space Θ as the expected value of V_f over P_θ, namely

$$J(\theta) = \mathbb{E}_{X\sim P_\theta}\left[V_f(X)\right] \quad .$$

Our algorithm performs the natural gradient descent on a Riemannian manifold $(\Theta, \mathcal{I}_\theta)$ equipped with the Fisher metric $\mathcal{I}_\theta$. The Fisher metric is the unique metric that does not depend on the choice of parameterization [3]. The natural gradient—the gradient taken w.r.t. the Fisher metric—is given by the product of the inverse of the Fisher information matrix $\mathcal{I}_\theta$ and the "vanilla" gradient $\nabla J(\theta)$ of the function. Therefore, the natural gradient of J is $\mathcal{I}_\theta^{-1}\nabla J(\theta)$ and the parameter update follows

$$\theta^{t+1} = \theta^t - \eta^t \mathcal{I}_{\theta^t}^{-1} \nabla J(\theta^t), \tag{1}$$

where η^t is the learning rate.

2.1 Deterministic NGD Algorithm on $\mathbb{R}^d$

In the following, we focus on the optimization in $\mathbb{R}^d$. Thus, $\mathbb{X} = \mathbb{R}^d$, μ is the Lebesgue measure μ_{Leb} on $\mathbb{R}^d$, and $\mathcal{F}$ is the Borel σ-field $\mathcal{B}^d$ on $\mathbb{R}^d$.

We choose as the sampling distribution the Gaussian P_θ parameterized by $\theta \in \Theta$, where the mean vector $m(\theta)$ is in $\mathbb{R}^d$ and the covariance matrix $C(\theta)$ is a symmetric and positive definite matrix of dimension d.

We define the invariant cost $V_f(x)$ by using the Lebesgue measure μ_{Leb} as $V_f(x) = \mu_{\mathrm{Leb}}^{2/d}[y : f(y) \leqslant f(x)]$. Then, the infimum of $J(\theta) = \mathbb{E}_{X\sim P_\theta}\left[V_f(X)\right]$ is zero located on a boundary of the domain Θ where the mean vector equals the global minimum of f and the covariance matrix is zero.

The choice of the parameterization of Gaussian distributions affects the behavior of the natural gradient update (1) with finite learning rate η^t, although the steepest direction of J on the statistical manifold Θ is invariant under the choice of parameterization. We choose the mean vector and the covariance matrix of the Gaussian distribution as the parameter as well as are chosen in the CMA-ES and in other algorithms such as EMNA [18] and cross entropy method [10]. Let $\theta = [m^{\mathrm{T}}, \mathrm{vec}(C)^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathrm{vec}(C)$ be the vectorization of C such that the (i, j)th element of C corresponds to $i + d(j-1)$th element of $\mathrm{vec}(C)$ (see [16]). Then the Fisher information matrix has an analytical form

$$\mathcal{I}_\theta = \begin{bmatrix} C^{-1} & 0 \\ 0 & \frac{1}{2}(C^{-1} \otimes C^{-1}) \end{bmatrix}, \tag{2}$$

where $\otimes$ denotes the Kronecker product operator. Under some regularity conditions for the exchange of integration and differentiation we have

$$\nabla J(\theta) = \mathbb{E}_{X\sim P_\theta}\left[V_f(X)\nabla l(\theta; X)\right], \tag{3}$$

where $l(\theta; x) = \ln p_\theta(x)$ is the log-likelihood. The gradient of the log-likelihood $\nabla l(\theta; x)$ can be written as

$$\nabla l(\theta; x) = \begin{bmatrix} C^{-1}(x-m) \\ \frac{1}{2}\,\mathrm{vec}(C^{-1}(x-m)(x-m)^{\mathrm{T}}C^{-1} - C^{-1}) \end{bmatrix}. \tag{4}$$

Then, the natural gradient $\mathcal{I}_\theta^{-1}\nabla J(\theta) = [\delta m^{\mathrm{T}}, \mathrm{vec}(\delta C)^{\mathrm{T}}]^{\mathrm{T}}$ at $\theta = \theta^t$ can be written by part

$$\delta m^t = \mathbb{E}_{X\sim P_{\theta^t}}[V_f(X)(X - m^t)]$$
$$\delta C^t = \mathbb{E}_{X\sim P_{\theta^t}}\left[V_f(X)\big((X - m^t)(X - m^t)^{\mathrm{T}} - C^t\big)\right]$$

With different learning rates for mean vector and covariance matrix updates, the natural gradient descent (1) reads

$$m^{t+1} = m^t - \eta_m^t \delta m^t, \quad C^{t+1} = C^t - \eta_C^t \delta C^t \quad . \tag{5}$$

We refer to (5) for the deterministic natural gradient descent (NGD) method.

2.2 Stochastic NGD Algorithm on $\mathbb{R}^d$

When $\nabla J(\theta)$ is not given, we need to estimate the gradient. We approximate the natural gradient and simulate the natural gradient descent as follows. Initialize the mean

vector m^0 and the covariance matrix C^0 and repeat the following steps until some termination criterion is satisfied:

1. Compute the eigenvalue decomposition of C^t, $[B, D] = \mathrm{eig}(C^t)$, where B is an orthogonal matrix and D is a diagonal matrix such that $C^t = BDB^{\mathrm{T}}$.
2. Compute the square root of C^t, $\sqrt{C^t} = B\sqrt{D}B^{\mathrm{T}}$.
3. Generate normal random vectors $z_1, \ldots, z_n \sim \mathcal{N}(0, I_d)$.
4. Compute $x_i = m^t + \sqrt{C^t} z_i$, for $i = 1, \ldots, n$.
5. Evaluate the objective values $f(x_1), \ldots, f(x_n)$;
6. Estimate $V_f(x_i)$ as
$$\widehat{V_f}(x_i) = \frac{(2\pi)^{d/2} \det(D)^{1/2}}{n} \sum_{j: f(x_j) \leqslant f(x_i)} \exp\left(\frac{\|z_j\|^2}{2}\right).$$
7. Compute the baseline $b = \sum_{i=1}^n \widehat{V_f}(x_i)/n$.
8. Compute the weights $w_i = (\widehat{V_f}(x_i) - b)/n$.
9. Estimate the natural gradient δm^t and δC^t as
$$\begin{aligned} \widehat{\delta m^t} &= \sum_{i=1}^n w_i (x_i - m^t) \\ \widehat{\delta C^t} &= \sum_{i=1}^n w_i \left((x_i - m^t)(x_i - m^t)^{\mathrm{T}} - C^t \right) . \end{aligned} \tag{6}$$
10. Compute the learning rates η_m^t and η_C^t.
11. Update the parameters as $m^{t+1} = m^t - \eta_m \widehat{\delta m^t}$ and $C^{t+1} = C^t - \eta_C \widehat{\delta C^t}$.

We refer to this algorithm for the stochastic NGD algorithm.

This algorithm generates n samples x_i from $\mathcal{N}(m^t, C^t)$ in steps 1–4 and evaluates their objective values in step 5. In step 6, the invariant costs $V_f(x_i)$ are evaluated. The estimates $\widehat{V_f}(x_i)$ are obtained as follows. By definition we have
$$V_f(x) = \left(\int \frac{\mathbf{1}_{\{f(y) \leqslant f(x)\}}}{p_{\theta^t}(y)} p_{\theta^t}(y) \mathrm{d}y \right)^{2/d}.$$
Applying Monte-Carlo approximation we have
$$\widehat{V_f}(x) = \left(\frac{1}{n} \sum_{j=1}^n \frac{\mathbf{1}_{\{f(x_j) \leqslant f(x)\}}}{p_{\theta^t}(x_j)} \right)^{2/d}. \tag{7}$$
Since $p_{\theta^t}(x_j) = \left((2\pi)^d \det(D)\right)^{-1/2} \exp\left(\|z_j\|^2/2\right)$, we have the estimates $\widehat{V_f}(x_i)$ in step 6. Step 7 computes the baseline b that is often introduced to reduce the estimation variance of gradients while adding no bias [12]. We simply choose the mean value of the $\widehat{V_f}(x_i)$ as the baseline. Replacing the expectation in (5) with the sample mean and adding the baseline (in step 8) we have the Monte-Carlo estimate of the natural gradient in step 9. Finally in step 11, we update the parameters along the estimated natural gradient with the learning rates computed in step 10. The learning rates are chosen in the following so that they are inverse proportional to the largest eigenvalue of the following matrix
$$Z^t = (C^t)^{-1/2} \widehat{\delta C^t} (C^t)^{-1/2} = \sum_{i=1}^n w_i (z_i z_i^{\mathrm{T}} - I_d) . \tag{8}$$

2.3 Difference from the IGO

The difference between the IGO algorithm and our deterministic algorithm is that the invariant cost in the IGO algorithm is defined by negative of the weighted quantile, $-w(P_{\theta^t}[y : f(y) \leqslant f(x)])$, where $w : [0, 1] \mapsto \mathbb{R}$ is non-increasing weight function. Since the quantile $P_{\theta^t}[y : f(y) \leqslant f(x)]$ depends on the current parameter θ^t, $V_f(x)$ for each x in the IGO algorithm changes from iteration to iteration, whereas it is fixed in our algorithm. This property makes our algorithm easier to analyze mathematically.

The difference between our stochastic algorithm and the pure rank-μ update CMA-ES [14] is the same as the difference between the deterministic one and the IGO algorithm. The pure rank-μ update CMA-ES approximates the quantile value $P_{\theta^t}[y : f(y) \leqslant f(x)]$ by the number of better solutions divided by the number of samples n, $R_i/n = |\{x_j : f(x_j) \leqslant f(x_i)\}|/n$. Therefore, the pure rank-μ update CMA-ES simulates the same lines as the stochastic NGD algorithm described in Section 2.2 with the weights $w_i = w(R_i/n)/n$.

In Section 4 we compare the stochastic NGD algorithm with the pure rank-μ update CMA-ES where
$$w_i = \begin{cases} 1/\lfloor n/4 \rfloor & \text{if } R_i/n \leqslant \lfloor n/4 \rfloor \\ 0 & \text{otherwize.} \end{cases} \tag{9}$$

2.4 Basic Properties

Invariance. Our algorithms inherit two important invariance properties from the IGO and the CMA-ES: invariance under monotonic transformation of the objective function and invariance under affine transformation of the search space (with the same transformation of the initial parameters). Invariance under monotonic transformation of the objective function makes the algorithm perform equally on a function f and on any composite function $g \circ f$ where g is any strictly increasing function. For example, the convex sphere function $f(x) = \|x\|^2$ is equivalent to the non-convex function $f(x) = \|x\|^{1/2}$ for this algorithm, whereas conventional gradient methods, e.g. Newton method, assume the convexity of the objective function and require a fine line search to solve non-convex functions. This invariance property is obtained as a result of the transformation $f \mapsto V_f$. Invariance under affine transformation of the search space is the essence of variable metric methods such as Newton's method. By adapting the covariance matrix, this algorithm attains universal performance on ill-conditioned objective functions.

Positivity. The covariance matrix of the Gaussian distribution must be positive definite and symmetric at each iteration. The next proposition gives the condition on the learning rate η_C^t such that the covariance matrix is always positive definite symmetric.

PROPOSITION 1. *Suppose that the learning rate for the covariance update $\eta_C^t < \lambda_1^{-1}(\sqrt{C^t}^{-1} \delta C^t \sqrt{C^t}^{-1})$ in the deterministic NGD algorithm, where $\lambda_1(\cdot)$ denotes the largest eigenvalue of the argument matrix. If C^0 is positive definite symmetric, C^t is positive definite symmetric for each t. Similarly, if $\eta_C^t < \lambda_1^{-1}(Z^t)$ in the stochastic NGD algorithm, where Z^t is defined in (8), and if C^0 is positive definite symmetric, the same result holds.*

PROOF. Consider the deterministic case (5). Suppose

that C^t is positive definite and symmetric. Then,

$$C^{t+1} = \sqrt{C^t}\left(I_d - \eta_C^t \sqrt{C^t}^{-1}\delta C^t \sqrt{C^t}^{-1}\right)\sqrt{C^t}.$$

Since $\eta_C^t < \lambda_1^{-1}(\sqrt{C^t}^{-1}\delta C^t\sqrt{C^t}^{-1})$ by the assumption, all the eigenvalues of $\eta_C^t\sqrt{C^t}^{-1}\delta C^t\sqrt{C^t}^{-1}$ is smaller than one. Thus, the inside of the brackets is positive definite symmetric and hence C^{t+1} is positive definite symmetric. By mathematical induction, we have that C^t is positive definite and symmetric for all $t \geqslant 0$. The analogous result for the stochastic case is obtained in the same way. □

Consistency. The gradient estimator (6) is not necessarily unbiased, yet it is consistent as is shown in the following proposition. Therefore, one can expect that the stochastic NGD approximates the deterministic NGD well when the sample size n is large. Let $\tilde{\nabla} : J \mapsto \mathcal{I}_\theta^{-1}\nabla J$ be the natural gradient operator.

PROPOSITION 2. *Let $X_1, \ldots, X_n$ be independent and identically distributed random vectors following P_θ. Let $\widehat{V_f}(x)$ and $G_\theta^n = [(\widehat{\delta m^t})^{\mathrm{T}}, \mathrm{vec}(\widehat{\delta C^t})^{\mathrm{T}}]^{\mathrm{T}}$ be the invariant cost (7) and the natural gradient (6) where $x_1, \ldots, x_n$ are replaced with $X_1, \ldots, X_n$. Suppose that*

$$\mathbb{E}[V_f(X)^2] < \infty. \tag{10}$$

Then, $G_\theta^n \overset{\text{a.s.}}{\to} \tilde{\nabla}J(\theta)$, where $\overset{\text{a.s.}}{\to}$ represents almost sure convergence.

PROOF. By the Cauchy-Schwarz inequality we have that $\mathbb{E}[\|V_f(X)\tilde{\nabla}l(\theta;X)\|]^2 < \mathbb{E}[V_f(X)^2]\mathbb{E}[\|\tilde{\nabla}l(\theta;X)\|^2]$. Note that $\mathbb{E}[\|\tilde{\nabla}l(\theta;X)\|^2] = \mathrm{Tr}(\mathcal{I}_\theta^{-1}) < \infty$. By Jensen's inequality we have that $\mathbb{E}[V_f(X)]^2 \leqslant \mathbb{E}[V_f(X)^2]$. Therefore, (10) implies

$$\mathbb{E}[\|V_f(X)\tilde{\nabla}l(\theta;X)\|] < \infty \text{ and} \tag{11}$$

$$\mathbb{E}[V_f(X)] < \infty\ . \tag{12}$$

Define $h_n(x) = \widehat{V_f}(x) - V_f(x)$ and decompose G_θ^n as

$$G_\theta^n = \frac{1}{n}\sum_{i=1}^n V_f(X_i)\tilde{\nabla}l(\theta;X_i) + \frac{1}{n}\sum_{i=1}^n h_n(X_i)\tilde{\nabla}l(\theta;X_i) - \underbrace{\left(\frac{1}{n}\sum_{i=1}^n \widehat{V_f}(X_i)\right)}_{=b}\left(\frac{1}{n}\sum_{i=1}^n \tilde{\nabla}l(\theta;X_i)\right). \tag{13}$$

By (11) and the strong law of large numbers (LLN), the first summand converges to $\mathbb{E}[V_f(X)\tilde{\nabla}l(\theta;X)] = \tilde{\nabla}J(\theta)$ almost surely as $n \to \infty$. So we have to show that the second term and the third term of (13) converge almost surely to zero.

First, we show the following almost sure convergence

$$\frac{1}{n}\sum_{i=1}^n h_n(X_i) \overset{\text{a.s.}}{\to} 0 \quad \text{as } n \to \infty. \tag{14}$$

By the definition of $\widehat{V_f}(x)$, we have

$$\lim_{n\to\infty}\widehat{V_f}(x) = \left(\lim_{n\to\infty}\frac{1}{n}\sum_{j=1}^n \frac{\mathbf{1}_{f(X_j)\leqslant f(x)}}{p_\theta(X_j)}\right)^{2/d}$$

and since (12) implies $\mu_{\mathrm{Leb}}[y : f(y) \leqslant f(x)] < \infty$ almost everywhere, we have by LLN

$$= \mu_{\mathrm{Leb}}^{2/d}[y : f(y) \leqslant f(x)] = V_f(x)$$

almost surely and almost everywhere in x. This implies $h_n(x) \overset{\text{a.s.}}{\to} 0$ almost everywhere in x.

For $m \leqslant n$, we have

$$\begin{aligned}\lim_{n\to\infty}\left|\frac{1}{n}\sum_{i=1}^n h_n(X_i)\right| &\leqslant \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n \sup_{j\geqslant n}|h_j(X_i)| \\ &\leqslant \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n \sup_{j\geqslant m}|h_j(X_i)|.\end{aligned} \tag{15}$$

Since $h_n(x) \overset{\text{a.s.}}{\to} 0$ almost everywhere in x as $n \to \infty$, we have $\sup_{j\geqslant m}|h_j(x)| \overset{\text{a.s.}}{\to} 0$ almost everywhere in x as $m \to \infty$. By the Lebesgue's dominated convergence theorem we have $\mathbb{E}[\sup_{j\geqslant m}|h_j(X)|] \to 0$ as $m \to \infty$. Therefore, we have that $\mathbb{E}[\sup_{j\geqslant m}|h_j(X)|] < \infty$ for m large enough. Then, by applying LLN, we have that the right most side of (15) converges to $\mathbb{E}[\sup_{j\geqslant m}|h_j(X)|]$ as $n \to \infty$ and this expectation converges to 0 as $m \to \infty$. This ends the proof of (14).

Now we can obtain the almost sure convergence of the third term of (13) to zero. Indeed, the almost sure convergence (14) implies that the limit $\lim_{n\to\infty}\sum_{i=1}^n \widehat{V_f}(X_i)/n$ agrees with $\lim_{n\to\infty}\sum_{i=1}^n V_f(X_i)/n$ and we have from (12) and LLN that $\sum_{i=1}^n \widehat{V_f}(X_i)/n \overset{\text{a.s.}}{\to} \mathbb{E}[V_f(X)] < \infty$. Also, by LLN we have that $\sum_{i=1}^n \tilde{\nabla}l(\theta;X_i)/n \overset{\text{a.s.}}{\to} 0$ as $n \to \infty$. Therefore, the third term of (13) converges to zero almost surely.

To show the convergence of the second term of (13) to zero, we apply the Cauchy-Schwarz inequality to it and we have

$$\left|\sum_{i=1}^n \frac{h_n(X_i)\tilde{\nabla}l(\theta;X_i)}{n}\right|^2 \leqslant \sum_{i=1}^n \frac{h_n(X_i)^2}{n}\sum_{i=1}^n \frac{\|\tilde{\nabla}l(\theta;X_i)\|^2}{n}.$$

By LLN we have that the second term of the right hand side converges to $\mathbb{E}[\|\tilde{\nabla}l(\theta;X)\|^2] = \mathrm{Tr}(\mathcal{I}_\theta^{-1})$. So we have to prove that the first term on the right hand side converges to zero almost surely. The proof of this convergence is done in the same way as above with h_n^2 replacing h_n. □

We remark that (12) is the necessary and sufficient condition for $J(\theta)$ to exist and that (11) is a sufficient condition for the exchange of integral and differentiation used in (3). See e.g. [9, Theorem 16.8].

3. CONVERGENCE PROPERTIES OF THE DETERMINISTIC NGD ALGORITHM

We investigate the convergence properties of the deterministic NGD algorithm (5) on a monotonic convex-quadratic composite function $f(x) = g(x^{\mathrm{T}}Ax)$, where g is any strictly increasing function and A is a positive definite symmetric matrix.

PROPOSITION 3. *The natural gradient can be written as*

$$\mathcal{I}_\theta^{-1}\nabla J(\theta) \propto \begin{bmatrix} CAm \\ \mathrm{vec}(CAC)\end{bmatrix}\ .$$

PROOF. Since $\mu_{\mathrm{Leb}}[y : f(y) \leqslant f(x)]$ is equivalent to the volume of the ellipsoid $\{y : y^{\mathrm{T}}Ay \leqslant x^{\mathrm{T}}Ax\}$, we have that

$$\mu_{\mathrm{Leb}}[y : f(y) \leqslant f(x)] = \frac{2}{\det(A)}V_d(\sqrt{x^{\mathrm{T}}Ax}),$$

where $V_d(r)$ denotes the volume of the sphere with radius r and is proportional to r^d. Therefore $V_f(x) = \mu_{\mathrm{Leb}}^{2/d}[y :$

$f(y) \leqslant f(x)] \propto x^{\mathrm{T}} A x$. Since the proportionality constant is independent of x, we have

$$\begin{aligned} J(\theta) &= \mathbb{E}_{X\sim P_\theta}\left[V_f(X)\right] \\ &\propto \mathbb{E}_{X\sim P_\theta}\left[X^{\mathrm{T}} A X\right] = m^{\mathrm{T}} A m + \mathrm{Tr}(AC). \end{aligned}$$

Differentiating the both side of the above relation, we have

$$\nabla J(\theta) \propto \begin{bmatrix} 2Am \\ \mathrm{vec}(A) \end{bmatrix}.$$

Premultiplying by $\mathcal{I}_\theta^{-1}$, we obtain the intended result. □

Now the deterministic NGD algorithm on $g(x^{\mathrm{T}} A X)$ is implicitly written as

$$m^{t+1} = m^t - \eta_m^t \delta m^t, \qquad \delta m^t = cC^t A m^t \tag{16}$$

$$C^{t+1} = C^t - \eta_C^t \delta C^t, \qquad \delta C^t = cC^t A C^t, \tag{17}$$

where $c > 0$ is the proportionality constant appearing in the proof of Proposition 3.

In the following, we work on the following assumption: There are $\gamma_{m,\min} > 0$ and $\gamma_{C,\min} > 0$ such that

$$\gamma_{m,\min} \leqslant \eta_m^t \lambda_1((C^t)^{-1}\delta C^t) \leqslant 1, \tag{18}$$

$$\gamma_{C,\min} \leqslant \eta_C^t \lambda_1((C^t)^{-1}\delta C^t) \leqslant 1/2. \tag{19}$$

These assumptions are satisfied, for example, if η_m^t and η_C^t are set for each iteration so that $\eta_m^t = \eta_C^t = \alpha/\lambda_1((C^t)^{-1}\delta C^t)$. In this case the natural gradient can be considered to be normalized by $\lambda_1((C^t)^{-1}\delta C^t)$ and the pseudo-learning rate is α.

The next theorem states that the covariance matrix converges proportionally to the inverse of the Hessian matrix.

THEOREM 4. *Assume* (19). *The condition number of $C^t A$ converges to one with the rate of convergence*

$$\limsup_{t\to\infty} \frac{\mathrm{Cond}(C^{t+1}A) - 1}{\mathrm{Cond}(C^t A) - 1} \leqslant \frac{1 - 2\gamma_{C,\min}}{1 - \gamma_{C,\min}}. \tag{20}$$

Moreover, we have an upper bound

$$\mathrm{Cond}(C^t A) \leqslant 1 + \left(\frac{1 - 2\gamma_{C,\min}}{1 - \gamma_{C,\min}}\right)^t (\mathrm{Cond}(C^0 A) - 1). \tag{21}$$

If the limit $\gamma_{C,\lim} = \lim_{t\to\infty} \eta_C^t \lambda_1((C^t)^{-1}\delta C^t)$ exists, $\gamma_{C,\min}$ is replaced with $\gamma_{C,\lim}$ in (20).

PROOF. Since A is positive definite and symmetric, there exists the square root $\sqrt{A}$. Premultiplying and postmultiplying both side of covariance matrix update (17) by $\sqrt{cA}$, we have

$$c\sqrt{A}C^{t+1}\sqrt{A} = c\sqrt{A}C^t\sqrt{A} - \eta_C^t (c\sqrt{A}C^t\sqrt{A})^2.$$

Since $c\sqrt{A}C^t\sqrt{A}$ is positive definite and symmetric, there exists an eigenvalue decomposition $Q^t \Lambda^t (Q^t)^{\mathrm{T}}$, where the diagonal elements of $\Lambda^t = \mathrm{diag}(\lambda_1^t, \ldots, \lambda_d^t)$ are the eigenvalues of $c\sqrt{A}C^t\sqrt{A}$ and each column of Q^t is the eigenvector corresponding to each diagonal element of Λ^t. Then,

$$c\sqrt{A}C^{t+1}\sqrt{A} = Q^t \left(\Lambda^t - \eta_C^t (\Lambda^t)^2\right) (Q^t)^{\mathrm{T}}.$$

This means, Q^t also diagonalizes $c\sqrt{A}C^{t+1}\sqrt{A}$. By mathematical induction we have that an orthogonal matrix Q which diagonalizes $c\sqrt{A}C^0\sqrt{A}$ diagonalizes $c\sqrt{A}C^t\sqrt{A}$ for any $t \geqslant 0$ and we have

$$\Lambda^{t+1} = \Lambda^t - \eta_C^t (\Lambda^t)^2. \tag{22}$$

Next, we show that the condition number of Λ^t converses to 1 as $t \to \infty$. Remember $\delta C^t = cC^t A C^t$. We have $\lambda_1((C^t)^{-1}\delta C^t) = \lambda_1(cAC^t) = \lambda_1(c\sqrt{A}C^t\sqrt{A}) = \lambda_1(\Lambda^t)$. Then, by assumption (19) we have

$$\gamma_{C,\min} \leqslant \eta_C^t \lambda_1(\Lambda^t) \leqslant 1/2. \tag{23}$$

Moreover, since $\lambda_1(\Lambda^t) \geqslant \lambda_i^t$ for any i, we have $\eta_C^t(\lambda_i^t + \lambda_j^t) \leqslant 1$ for any i, j.

Suppose $\lambda_i^t \geqslant \lambda_j^t$ without loss of generality. From (22) and the inequality $\eta_C^t(\lambda_i^t + \lambda_j^t) \leqslant 1$, we have

$$\begin{aligned} \lambda_i^{t+1} - \lambda_j^{t+1} &= \lambda_i^t(1 - \eta_C^t\lambda_i^t) - \lambda_j^t(1 - \eta_C^t\lambda_j^t) \\ &= (1 - \underbrace{\eta_C^t(\lambda_i^t + \lambda_j^t)}_{\leqslant 1})\underbrace{(\lambda_i^t - \lambda_j^t)}_{\geqslant 0} \geqslant 0 \end{aligned}$$

with equality holding if and only if $\lambda_i^t = \lambda_j^t$. Therefore, if $\lambda_i^t > \lambda_j^t$, then $\lambda_i^{t+1} > \lambda_j^{t+1}$, which implies that if ith and jth diagonal elements of Λ^0 are the maximum and minimum elements, ith and jth elements of Λ^t are also the maximum and minimum elements of Λ^t. Without loss of generality we suppose $\lambda_i^t \geqslant \lambda_j^t$ for any $i \leqslant j$ for all $t \geqslant 0$. Then, $\lambda_1^t/\lambda_d^t = \mathrm{Cond}(\Lambda^t) = \mathrm{Cond}(C^t A)$. According to (22) we have

$$\begin{aligned} \underbrace{\frac{\lambda_1^{t+1} - \lambda_d^{t+1}}{\lambda_d^{t+1}}}_{\mathrm{Cond}(C^{t+1}A)-1} &= \frac{\lambda_1^t(1 - \eta_C^t\lambda_1^t) - \lambda_d^t(1 - \eta_C^t\lambda_d^t)}{\lambda_d^t(1 - \eta_C^t\lambda_d^t)} \\ &= \underbrace{\frac{(\lambda_1^t - \lambda_d^t)}{\lambda_d^t}}_{\mathrm{Cond}(C^tA)-1} \frac{1 - \eta_C^t(\lambda_1^t + \lambda_d^t)}{(1 - \eta_C^t\lambda_d^t)}. \end{aligned} \tag{24}$$

Since

$$\frac{1 - \eta_C^t(\lambda_1^t + \lambda_d^t)}{(1 - \eta_C^t\lambda_d^t)} = \frac{1 - \eta_C^t\lambda_1^t(1 + \lambda_d^t/\lambda_1^t)}{1 - \eta_C^t\lambda_1^t(\lambda_d^t/\lambda_1^t)} \leqslant \frac{1 - 2\eta_C^t\lambda_1^t}{1 - \eta_C^t\lambda_1^t},$$

we have

$$\frac{\mathrm{Cond}(C^{t+1}A) - 1}{\mathrm{Cond}(C^t A) - 1} \leqslant \frac{1 - 2\eta_C^t\lambda_1^t}{1 - \eta_C^t\lambda_1^t}. \tag{25}$$

Moreover, since the right-hand side of the above inequality is maximized when $\eta_C^t\lambda_1^t$ is minimized and $\eta_C^t\lambda_1^t$ is bounded from below by $\gamma_{C,\min}$ because of (23), we have

$$\frac{\mathrm{Cond}(C^{t+1}A) - 1}{\mathrm{Cond}(C^t A) - 1} \leqslant \frac{1 - 2\gamma_{C,\min}}{1 - \gamma_{C,\min}}.$$

This implies $\lim_{t\to\infty} \mathrm{Cond}(C^t A) = 1$. The rate of convergence (20) and the upper bound (21) are immediate consequences of the above inequality.

If the limit $\gamma_{C,\lim}$ exists, it is easy to see from (25) that $\gamma_{C,\min}$ can be replaced with $\gamma_{C,\lim}$ in (20). This completes the proof. □

Note that if $\eta_C^t = \alpha/\lambda_1((C^t)^{-1}\delta C^t)$ and $\alpha \leqslant 1/2$, we have that $\gamma_{C,\lim} = \gamma_{C,\min} = \alpha$. We have from (24) that

$$\mathrm{Cond}(C^{t+1}A) = \mathrm{Cond}(C^t A)\frac{1 - \alpha}{1 - \alpha\,\mathrm{Cond}^{-1}(C^t A)} \tag{26}$$

and the rate of convergence becomes $(1 - 2\alpha)/(1 - \alpha)$.

The next theorem states the global convergence of m and C and the speed of the convergence. In the following, we let $\|M\|$ denote the Frobenius norm of M, namely $\|M\| = \mathrm{Tr}^{1/2}(M^{\mathrm{T}} M)$.

THEOREM 5. *Assume* (19) *and* (18). *Then,* $\|m^t\|$ *and* $\|C^t\|$ *converge to zero with the rate of convergence*

$$\limsup \frac{\|\kappa^{t+1}\|}{\|\kappa^t\|} \leqslant 1 - \gamma_{\kappa,\min}, \tag{27}$$

where κ *is either* m *or* C *and* κ^t *is either* m^t *or* C^t. *If the limit* $\gamma_{\kappa,\lim} = \lim_{t\to\infty} \eta_\kappa^t \lambda_1((C^t)^{-1}\delta C^t)$ *exists,* $\gamma_{\kappa,\min}$ *is replaced with* $\gamma_{\kappa,\lim}$ *in* (27).

PROOF. Let $\sigma_i(\cdot)$ denote the ith largest singular value of the argument matrix. According to J. von Neumann's trace inequality [19] we have $|\mathrm{Tr}(M_1 M_2)| \leqslant \sum_{i=1}^d \sigma_i(M_1)\sigma_i(M_2) \leqslant \sigma_1(M_1)\sum_{i=1}^d \sigma_i(M_2)$, where M_1 and M_2 are any matrices in $\mathbb{R}^{d\times d}$. Let $M \in \mathbb{R}^{d\times d}$ be nonnegative definite and $S \in \mathbb{R}^{d\times d}$ is nonnegative definite symmetric. From the above inequality, we have

$$\begin{aligned}\|MS\|^2 &= \mathrm{Tr}(SM^{\mathrm{T}}MS) = \mathrm{Tr}(M^{\mathrm{T}}MS^2)\\ &\leqslant \sigma_1(M^{\mathrm{T}}M)\sum_{i=1}^d \sigma_i(S^2) = \sigma_1^2(M)\|S\|^2.\end{aligned}$$

Applying this matrix norm inequality and the vector norm inequality $\|Mx\|^2 \leqslant \sigma_1(M)^2\|x\|^2$ to (16) and (17), we have

$$\frac{\|\kappa^{t+1}\|}{\|\kappa^t\|} \leqslant \sigma_1(I - c\eta_\kappa^t C^t A).$$

In light of Theorem 4, we have that $\lim_{t\to\infty} C^t A/\lambda_1(C^t A) = I_d$. Then, from the assumptions (18) and (19) we have

$$\begin{aligned}&\limsup_t \sigma_1(I - c\eta_\kappa^t C^t A)\\ &= \limsup_t \sigma_1\Bigg(I - \underbrace{\eta_\kappa^t \lambda_1(cC^t A)}_{\geqslant \gamma_{\kappa,\min}} \underbrace{\frac{cC^t A}{\lambda_1(cC^t A)}}_{\to I_d}\Bigg) \leqslant 1 - \gamma_{\kappa,\min}.\end{aligned}$$

This implies linear convergence of κ with rate of convergence at most $1 - \gamma_{\kappa,\min}$.

If the limit $\gamma_{\kappa,\lim}$ exists, we can easily see from the above inequality that $\gamma_{\kappa,\min}$ is replaced with $\gamma_{\kappa,\lim}$ in (27). This ends the proof. □

Now we can see the importance of the covariance matrix adaptation quantitatively. Let $\eta_m^t = \eta_C^t = \alpha/\lambda_1((C^t)^{-1}\delta C^t)$. Then, the covariance matrix becomes proportional to the inverse of the Hessian at the speed given by (26) and the rate of convergence of the parameter becomes $1-\alpha$. Meanwhile, if the covariance matrix is restricted to a product of a scalar v^t and the identity matrix, $C^t = v^t I$, then the rate of convergence is in $[1-\alpha, 1-\alpha\,\mathrm{Cond}^{-1}(A)]$[1]. Therefore, the rate of convergence becomes close to one in the worst case if $\mathrm{Cond}(A) \gg 1$.

From Theorem 4 we know that the deterministic NGD algorithm learns the inverse of the Hessian. The convergence of the covariance matrix to the inverse of the Hessian matrix in the CMA-ES has been anticipated but it has not been proven. Theorem 4 demonstrates this anticipation affirmatively for the deterministic NGD algorithm. Theorem 5 exhibits the linear convergence of the parameters. This implies that the rate of convergence of the expected objective value $J(\theta) \propto m^{\mathrm{T}}Am + \mathrm{Tr}(CA)$ is also linear and equals to the rate of convergence of $\|C^t\|$.

[1]If the covariance matrix is restricted to a diagonal matrix, the target matrix is $\mathrm{diag}(A) = \mathrm{diag}(A_{1,1}, \ldots, A_{d,d})$, i.e. $\lim_t \mathrm{Cond}(C^t\,\mathrm{diag}(A)) = 1$. the rate of convergence is in $[\sigma_d(I - \alpha\,\mathrm{Cond}(\tilde{A})), \sigma_1(I - \alpha\,\mathrm{Cond}(\tilde{A}))]$, where $\tilde{A} = \mathrm{diag}(A)^{-1}A$. We omit the derivation due to the space limitation.

4. NUMERICAL SIMULATION

The results in the previous section are for the deterministic (ideal) NGD algorithm. Thanks to Proposition 2 we can expect that the stochastic NGD algorithm proposed in Section 2.2 approximates the deterministic one arbitrarily close as the sample size n is taken sufficiently large. In this section, we evaluate how well the stochastic variant with finite sample size approximates the deterministic one on a quadratic function.

We consider the 20-dimensional ellipsoid function

$$f(x) = \sum_{i=1}^d 10^{\frac{6(i-1)}{d-1}} x_i^2 \qquad (d = 20).$$

Note that the ellipsoid function is separable and convex but our algorithm does not exploit the separability and convexity. The eigenvalues (diagonal elements) of the Hessian matrix of the ellipsoid function range in $[1, 10^6]$. We set the initial parameters as $m^0 = (0, \ldots, 0)^{\mathrm{T}}$ and $C^0 = I_d$.

We design the learning rates as

$$\eta_m^t = \frac{1}{\sigma_1(Z^t)} \text{ and } \eta_C^t = \frac{c_C}{2\sigma_1(Z^t)}, \quad c_C \leqslant 1.$$

Here, Z^t is a matrix defined in (8).

4.1 Effect of Sample Size and Learning Rate

First, we investigate the effect of the sample size n and the coefficient c_C of the learning rate η_C^t. We try the following sample sizes: $\lceil d^{1/2}\rceil$, d, $\lceil d^{3/2}\rceil$, d^2, $\lceil d^{5/2}\rceil$, d^3.

Figure 1 illustrates the slope of the condition number $\mathrm{Cond}(C^t A)$ and the theoretical curve (26) and the slope of the expected objective function value $\mathbb{E}_{X\sim P_{\theta^t}}[X^{\mathrm{T}}AX] = (m^t)^{\mathrm{T}}Am^t + \mathrm{Tr}(C^t A)$, averaged over 50 independent trials. When the sample size is larger, we see the closer performance to the theoretical result. When $n = d^3$ and $c_C = 0.1$, the convergence curve of the condition number approximated well the theoretical curve and the final condition number is $\mathrm{Cond}(C^t A) \approx 1.1$. When $n = \lceil d^{1/2}\rceil$ and $c_C = 0.1$, it takes more than 30 times longer to learn the covariance matrix and the final condition number becomes $\mathrm{Cond}(C^t A) \approx 4.0$, although the stochastic algorithm still works successfully. We attain a little higher condition numbers when we choose larger learning rates $c_C = 0.5, 1.0$. For example, the final condition numbers are $\mathrm{Cond}(C^t A) \approx 1.3$ for $n = d^3$ and $c_C = 0.5$, and $\mathrm{Cond}(C^t A) \approx 1.6$ for $n = d^3$ and $c_C = 1.0$. This is because smaller learning rates have more effect of averaging the natural gradient estimates over iterations and reducing the estimation variance.

Note that we observe a slightly slower adaptation of the covariance matrix at the beginning in case that we set $m^0 = (10, \ldots, 10)$, although the adaptation behavior (26) does not change in theory. See Figure 2. This attributes to the estimation precision of $\widehat{V_f}$. If the squared Mahalanobis distance $(m^t)^{\mathrm{T}}(C^t)^{-1}m^t$ between the origin (the global optimum) and the current mean with respect to C^t is larger, the function landscape around m^t looks more like linear function. Then $\widehat{V_f}(x_i)$ are far from the exact values, especially in case a small sample size is chosen.

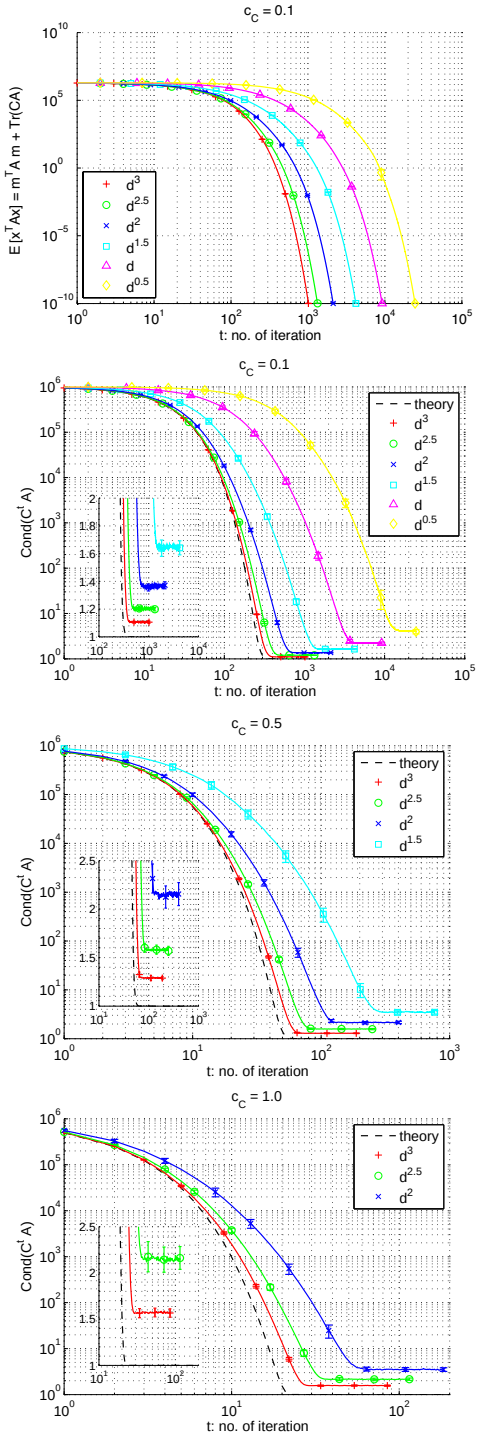

Figure 1: Averages and standard deviations of the condition numbers $\mathrm{Cond}(C^t A)$ for $c_C = 0.1$, 0.5, and 1.0 and the expected objective function values for $c_C = 0.1$. Theoretical curves (26) of the condition number are also illustrated with dashed lines. All the lines are cut after first reach of the expected objective value to 10^{-10}. Some results are omitted, for example $n = d$ for $c_C = 0.5$, because numerically unstable computation occurs during search.

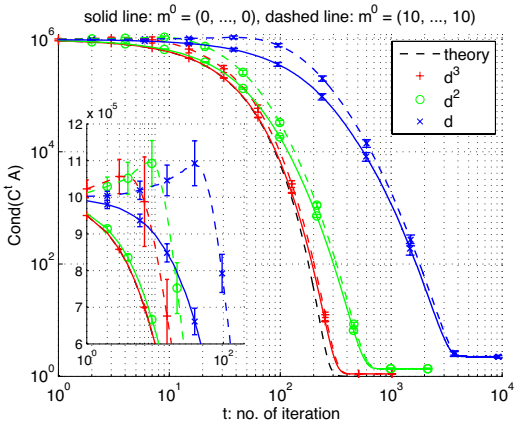

Figure 2: Averages and standard deviations of the change of the condition numbers for $c_C = 0.1$ for $n = d$, d^2, d^3 with initial mean vector $m = (0, \ldots, 0)$ or $m = (10, \ldots, 10)$. Other settings are the same as in Figure 1.

4.2 Comparison with Rank-μ update CMA-ES

Finally, we study how well this stochastic algorithm simulates the CMA-ES. We test the pure rank-μ update CMA-ES with weight scheme (9). We set the learning rates following [13]

$$\eta_m^t = 1, \qquad \eta_C^t = \frac{2\mu_w - 1}{(d+2)^2 + \mu_w},$$

where $\mu_w = 1/\sum_{i=1}^n w_i^2$. We choose c_C for our algorithm so that the speed of adaptation for each model is almost the same.

Figure 3 shows the results for each method for $n = d$ and $n = d^2$. In both case, we confirm similar behaviors of the pure rank-μ update CMA-ES and our algorithm despite their dissimilar weight-value settings. The similar change of performance illustrated in Figure 2 is also observed for the pure rank-μ update CMA-ES. From this result, we conclude that it is possible to estimate the performance of the pure rank-μ update CMA-ES by our natural gradient algorithm, which is theoretically more attractive.

However, note that the pure rank-μ update CMA-ES is not the standard CMA-ES [13] and the standard CMA-ES performs better than the pure rank-μ update CMA-ES. The standard CMA-ES employes so-called evolution paths to adapt the covariance matrix and the global scale of the covariance matrix, which is called step-size in the CMA-ES context. Moreover, the standard CMA-ES employes weighted recombination, where different values are assigned to the weights for $R_i \leqslant \lfloor n/2 \rfloor$, which is only slightly better than intermediate recombination (9) and even similar to our setting. Furthermore, the similar performance observed is only on a quadratic function. If there are certain functions which distinguish our algorithm from the (rank-μ) CMA-ES, this may help to understand both the NGD algorithm and the CMA-ES. Further study on these topics is required.

5. CONCLUSION

We have proposed a novel natural gradient descent (NGD) method where the objective function is transformed to a function defined on the parameter space of probability dis-

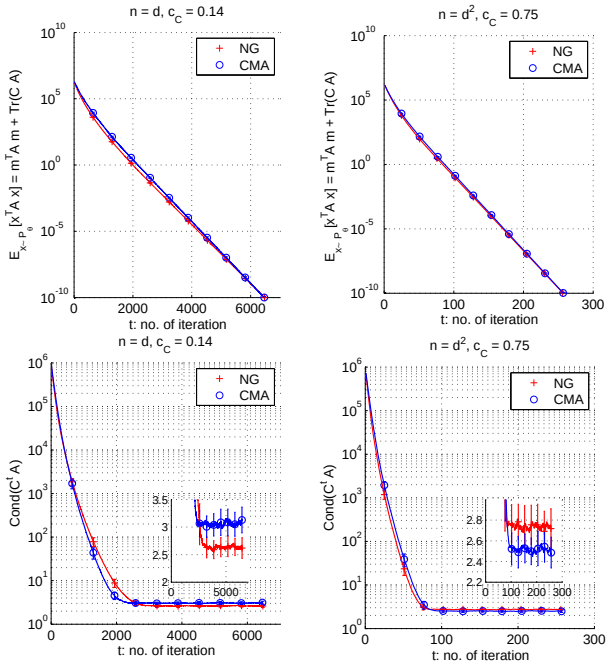

Figure 3: Averages and standard deviations of the change of the condition numbers and the change of the expected objective function values for $n = d$ and $c_C = 0.14$ on the left, and for $n = d^2$ and $c_C = 0.75$ on the right. Other settings are the same as in Figure 1.

tributions. We have proven that the deterministic NGD method learns the inverse of the Hessian of the original objective function that is any monotonic convex-quadratic-composite function. Linear convergence of the mean vector and the covariance matrix has been also proven. The numerical results for the stochastic NGD algorithm have shown that the stochastic algorithm approximates the deterministic one well when the sample size is sufficiently large. Moreover, we have confirmed that the stochastic NGD algorithm and the pure rank-μ update CMA-ES behave very similarly on a quadratic function.

The contribution of the paper is to derive a novel NGD algorithm that can be viewed as a variant of the CMA-ES from the first principle of information geometry. This allows us to analyze the algorithm theoretically. Our theoretical results in Section 3 imply that there is at least one weight-value setting in the CMA-ES such that the covariance matrix learns the inverse of the Hessian of the objective function. Moreover, since our algorithm does not only share most of the important properties of the rank-μ update CMA-ES, but also is confirmed to perform similarly to the pure rank-μ update CMA-ES on a quadratic function by numerical simulations, we could study our algorithm to find out limitations of the pure rank-μ update CMA-ES and to discover a way to improve the CMA-ES.

6. REFERENCES

[1] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Bidirectional Relation between CMA Evolution Strategies and Natural Evolution Strategies. In *Parallel Problem Solving from Nature – PPSN XI*, pages 154–163, 2010.

[2] S. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998.

[3] S. Amari and H. Nagaoka. *Methods of Information Geometry*. Methods of Information Geometry. American Mathematical Society, 2007.

[4] S. Amari, H. Park, and K. Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12:1399–1409, 2000.

[5] L. Arnold, A. Auger, N. Hansen, and Y. Ollivier. Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. *arXiv:1106.3708v1*, 2011.

[6] A. Auger. Convergence results for the (1, λ)-SA-ES using the theory of φ-irreducible Markov chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[7] S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In *Proceedings of the 12th International Conference on Machine Learning*, pages 38–46, 1995.

[8] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[9] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience, third edition, 1995.

[10] P.-T. D. Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A Tutorial on the Cross-Entropy Method. *Annals of Operations Research*, pages 19–67, 2005.

[11] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 393–400. ACM, 2010.

[12] E. Greensmith, P. L. Bartlett, and J. Baxter. Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

[13] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 282–291, 2004.

[14] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[15] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[16] D. A. Harville. *Matrix Algebra from a Statistician's Perspective*, Springer Verlag, 2008.

[17] J. Jägersküpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007.

[18] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer Netherlands, 2002.

[19] L. Mirsky. A trace inequality of John von Neumann. *Monatshefte für Mathematik*, 79(4):303–306, 1975.

[20] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, second edition, 2006.

[21] H. Park, S. Amari, and K. Fukumizu. Adaptive natural gradient learning algorithms for various stochastic models. *Neural networks : the official journal of the International Neural Network Society*, 13(7):755–764, 2000.

[22] J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[23] M. Rattray, D. Saad, and S. Amari. Natural gradient descent for on-line learning. *Physical review letters*, 81(24):5461–5464, 1998.